1. **项目系统总体架构**



图1-1 系统总体架构图

1. 如图1-1所示，原动力负责开发的内容为虚线框内的通信链路；本系统采用LORA跳频扩频通信方式，支持双向点对点通信，即节点至网关、网关至节点，可用于远程控制及数据采集，且已有校验和纠错功能；暂不支持丢包重传，可由用户自行实现。
2. 开发内容主要分为两部分：网关以及设备节点。
3. 设备节点的主要功能为等待【二次开发MCU】上传的数据信息并保存，等待网关发来读取指令时返回数据信息。
4. 网关负责管理底下的设备节点，采用轮询的方式收集节点的数据，收集到数据后主动发送给【二次开发MUC】；接收【二次开发MUC】发来的数据，并将其发送至节点。
5. 原动力开发的通信链路整体做成透传系统，即【二次开发MCU】上传的数据在网关轮询到了之后即刻主动发送给上层的【二次开发MCU】，没有轮询到数据即不发送。

（PS:做成透传系统方便二次开发）

1. **系统通信层次结构图**



图2-1 通信分层图示

1. 如图2-1为系统各层间的通信方式呈现图；
2. 二次开发MCU获取数据的方式是由网关主动发送给它，通信方式为串口通信；
3. 网关与设备节点的通信方式为Lora通信，由网关主动轮询设备节点读回数据；
4. 设备节点与二次开发MCU为串口通信，由二次开发MCU主动发送数据给设备节点；
5. **网关和设备节点通信**
6. **概述**
7. 每个网关和设备节点都有一个12字节（96位）的唯一硬件地址，分别称为GMAC地址和NMAC地址；
8. 网关包含至少两个Lora模块，作用如下：

|  |  |
| --- | --- |
| Lora模块A | Lora模块B、C、D... |
| 1.始终使用【配置参数1】于公共信道侦听设备节点的【入网请求包】及发送【入网授权包】。2.在配置模式下，侦听设备节点发出的【绑定请求包】及发送【绑定授权包】 | 1.发送【数据查询包】查询设备节点数据。2.侦听【数据返回包】收集设备节点数据 |

1. 设备节点包含一个Lora模块；
2. 设备节点只能与事先绑定的网关通信，设备节点绑定网关实质是记录网关的GMAC地址；
3. 设备节点可通过按下reset按键清除与网关的绑定关系（清除记录的GMAC地址）；
4. 网关维护一个【在线设备节点列表】，并按顺序轮询列表中的设备节点，收集它们的数据；
5. **网关和设备节点绑定**

图3-2-1 网关与设备节点绑定流程图

1. 按下网关的配置按键使网关进入配置模式，同时网关将侦听设备节点发出的【绑定请求包】；
2. 按下设备节点的reset按键（实质是清除自身记录的GMAC地址）使设备节点进入主动请求绑定的状态（若设备节点是第一次上电由于没有GMAC地址所以会自动进入主动请求绑定的状态）；
3. 设备节点使用【配置参数1】在公共信道上发出包含自身NMAC地址的【绑定请求包】；
4. 处在配置模式的网关接收到【绑定请求包】后会进行下一步的处理，没有处在配置模式的网关将忽略该包；
5. 配置模式下的网关收到【绑定请求包】将会给设备节点回复【绑定授权包】，【绑定授权包】包含请求设备节点的NMAC地址和网关自身的GMAC地址；
6. 设备节点收到【绑定授权包】后判断【绑定授权包】中的NMAC地址是否是自身的NMAC地址；
7. 如果是则会获取并记录【绑定授权包】中的GMAC地址完成绑定，否则忽略此包并继续尝试绑定；
8. 网关在配置模式下被按下配置按键将退出配置模式；
9. 设备节点发出【绑定请求包】后在【时间参数2】时间内没有收【绑定授权包】则会延时【时间参数1】后再发送【绑定请求包】。
10. **设备节点入网**

图3-3-1 设备节点入网流程图

1. 设备节点上电首先检查自身是否记录了GMAC地址，若没有则请求绑定的流程；若已有GMAC地址则发出包含自身NMAC地址的【入网请求包】；
2. 网关接收到【入网请求包】后判断已入网的设备数是否到达【入网上限数】；
3. 若到达则回复【入网拒绝包】；
4. 若没有到达则回复【入网授权包】，并且在【在线设备节点列表】中添加一台新的设备节点同时记录分配给该设备的【配置参数2】并标志其在线；
5. 【入网拒绝包】和【入网授权包】都包含请求设备节点的NMAC地址，且【入网授权包】还包含了分配给设备节点的【配置参数2】，即通信频点、带宽、扩频因子及纠错率；
6. 设备节点收到【入网授权包】后判断【入网授权包】中的NMAC地址是否为自身的NMAC地址，如果不是则会忽略该【入网授权包】，如果是则会记录【入网授权包】中的【配置参数2】，并标记自身为在线状态，同时采用【配置参数2】侦听网络；
7. 若设备节点收到【入网拒绝包】后会延时【时间参数1】后重新发出【入网请求包】；
8. 如果超过【时间参数2】设备节点没有接受到【入网拒绝包】或【入网授权包】会延时【时间参数1】后重新发出【入网请求包】；
9. **网关和设备节点正常通信**

图3-4-1 网关与设备节点通信逻辑图

1. 网关按【在线设备节点列表】中的顺序轮询设备节点；
2. 每轮询到一个设备节点就采用【配置参数2】向该设备节点发送【数据查询包】，发送完成后即可采用【配置参数2】侦听网络；
3. 设备节点入网成功后使用收到的【入网授权包】中的【配置参数2】侦听网络；
4. 设备节点收到来自网关的【数据查询包】时，即刻采用【配置参数2】发送【数据返回包】给网关；
5. 网关接收到【数据返回包】后解析包中的数据，并把数据发送给【二次开发MCU】完成本次轮询进入下一个设备节点的轮询；
6. 网关发送完成【数据查询包】等待【时间参数2】后没有收到【数据返回包】则记录该设备节点超时次数；紧接着判断该设备节点是否连续超时三次；如果是则标志该设备节点离线，该设备节点需要重新申请入网才会被轮询到，然后返回轮询下一个设备节点；如果不是则直接返回去轮询下一个设备节点；
7. 设备节点从收到【入网授权包】或上一次【数据查询包】等待【时间参数3】后没有收到【数据查询包】则认为自己离线了，并延时【时间参数1】后请求入网；
8. **关键词的解释**
9. **二次开发MCU**
10. 原动力开发的网关和设备节点都做成添加了邮票孔的PCB板，作为一个贴片器件贴在二次开发的PCB板上；
11. 网关和设备节点与二次开发MCU均采用串口通信的方式通信；
12. 设备网关及设备节点有一个RX\_EN\_N引脚，二次开发MCU可以通过该IO是否为低电平，判断当前是否可以发送数据；
13. RX\_EN\_N为高电平时，网关和设备节点收到的数据将被忽略。
14. 数据由二次开发MCU主动发送给设备节点，网关逐个轮询设备节点将数据逐条发送给上层的二次开发MCU，没有轮询到数据则不发送（即原动力开发的链路系统为透传系统）；
15. 网关和设备节点数据到二次开发MCU数据的格式，如下图所示：

|  |  |  |
| --- | --- | --- |
| 12byte | 12byte | n byte |
| 源地址 | 本机地址 | 用户数据 |

1. 二次开发MCU发送数据到网关和设备节点时带上目标设备的MAC地址，数据帧如下图所示：（**注意：对于设备节点，当目标地址全部为0x00时，数据将发送至网关；目标地址全部为0xff时，将使用【快速数据帧】并在公共信道广播传输**）

|  |  |
| --- | --- |
| 12byte | n byte |
| 目标地址 | 用户数据 |

1. 串口指令

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 读取配置参数1 | 指令 | 无参 |  |  |  |  |
| 0x01 | 无参 |  |  |  |  |
| 回复读取配置参数1 | 指令 | 4byte | 4bit | 4bit | 3bit | 5bit |
| 0x02 | FHKEY | SF | SB | EC | reserved |
| 读取配置参数2 | 指令 | 无参 |  |  |  |  |
| 0x03 | 无参 |  |  |  |  |
| 回复读取配置参数2 | 指令 | 4bit | 4bit | 3bit | 5bit |  |
| 0x04 | SF | SB | EC | reserved |  |
| 读取上限节点数 | 指令 | 无参 |  |  |  |  |
| 0x05 | 无参 |  |  |  |  |
| 回复读取上限节点数 | 指令 | 1byte |  |  |  |  |
| 0x06 | number |  |  |  |  |
| 读取时间 | 指令 | 无参 |  |  |  |  |
| 0x07 | 无参 |  |  |  |  |
| 回复读取时间 | 指令 | 1byte | 1byte | 1byte |  |  |
| 0x08 | time1 | time2 | time3 |  |  |
| 读取状态 | 指令 | 无参 |  |  |  |  |
| 0x09 | 无参 |  |  |  |  |
| 回复读取状态(网关) | 指令 | 12byte | 4byte | 2byte | 1byte |  |
| 0x0a | NMACn | FHKEYn | RSSIn | SNRn |  |
| 回复读取状态(节点) | 指令 | 12byte | 1byte |  |  |  |
| 0x0a | NMAC | status |  |  |  |
| 写入配置参数1 | 指令 | 4byte | 4bit | 4bit | 3bit | 5bit |
| 0x0b | FHKEY | SF | SB | EC | reserved |
| 回复写入配置参数1 | 指令 | 无参 |  |  |  |  |
| 0x0c | 无参 |  |  |  |  |
| 写入配置参数2 | 指令 | 4bit | 4bit | 3bit | 5bit |  |
| 0x0e | SF | SB | EC | reserved |  |
| 回复写入配置参数2 | 指令 | 无参 |  |  |  |  |
| 0x0f | 无参 |  |  |  |  |
| 写入上限节点数 | 指令 | 1byte |  |  |  |  |
| 0x10 | number |  |  |  |  |
| 回复写入上限节点数 | 指令 | 无参 |  |  |  |  |
| 0x11 | 无参 |  |  |  |  |
| 写入时间 | 指令 | 1byte | 1byte | 1byte |  |  |
| 0x12 | time1 | time2 | time3 |  |  |
| 回复写入时间 | 指令 | 无参 |  |  |  |  |
| 0x13 | 无参 |  |  |  |  |

1. 二次开发MCU通过CMD\_EN\_N的低电平使能通知原动力开发的网关或设备节点接下来要发送的是指令；
2. **配置参数1**

|  |  |  |  |
| --- | --- | --- | --- |
| FHKEY | SpreadingFactor | SignalBw | ErrorCoding |
| 0 | 10 | 500 kHz | 4/8 |

1. 配置参数1为公共信道使用的配置参数；
2. 网关与设备节点的绑定以及设备节点入网请求均采用该信道通信；
3. **配置参数2**

|  |  |  |  |
| --- | --- | --- | --- |
| FHKEY | SpreadingFactor | SignalBw | ErrorCoding |
| N | 9 | 250 kHz | 4/8 |

1. 配置参数2是设备节点请求入网时网关分配给设备节点的配置参数；
2. 每个设备节点被分配得到的配置参数都不一样；
3. N为随机数，网关和设备节点都会记录该随机数；
4. 正常通信时双方都会采用该随机数产生数字序列用于跳频通信；
5. **绑定请求包**

|  |  |
| --- | --- |
| 1byte | 12byte |
| 【Flag】 | NMAC |

1. 【数据包类型】：0，【数据包方向】：1；
2. 数据包长度：13字节；
3. NMAC：设备节点的唯一ID。
4. **绑定授权包**

|  |  |  |
| --- | --- | --- |
| 1byte | 12byte | 12byte |
| 【Flag】 | NMAC | GMAC |

1. 【数据包类型】：0，【数据包方向】：0；
2. 数据包长度：25字节；
3. NMAC：设备节点的唯一ID；
4. GMAC：网关的唯一ID。
5. **入网请求包**

|  |  |  |
| --- | --- | --- |
| 1byte | 12byte | 12byte |
| 【Flag】 | NMAC | GMAC |

1. 【数据包类型】：1，【数据包方向】：1；
2. 数据包长度：25字节；
3. NMAC：设备节点的唯一ID；
4. GMAC：网关的唯一ID。
5. **入网授权包**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1byte | 12byte | 4byte | 4bit | 4bit | 3bit | 5bit |
| 【Flag】 | NMAC | FHKEY | SF | SB | EC | reserved |

1. 【数据包类型】：1，【数据包方向】：0；
2. 数据包长度：19字节；
3. NMAC：设备节点的唯一ID；
4. FHKEY：随机数，用于产生跳频序列；
5. SF：扩频因子；
6. SB：带宽；
7. EC：纠错率；
8. Reserved：
9. **数据查询包**

|  |  |
| --- | --- |
| 1byte | N byte |
| 【Flag】 | data |

1. 【数据包类型】：2，【数据包方向】：0；
2. 数据包长度：1+N字节；
3. Date:用户发送的数据；
4. **数据返回包**

|  |  |
| --- | --- |
| 1byte | N byte |
| 【Flag】 | data |

1. 【数据包类型】：2，【数据包方向】：1；
2. 数据包长度：1+N字节；
3. Date:用户发送的数据；
4. **快速数据帧**

|  |  |  |
| --- | --- | --- |
| 1byte | 12byte | N byte |
| 【Flag】 | NMAC | data |

1. 【数据包类型】：3，【数据包方向】：1；
2. 数据包长度：1+N字节；
3. NMAC：设备节点的唯一ID；
4. Date:用户发送的数据；
5. **在线设备节点列表**

|  |  |  |  |
| --- | --- | --- | --- |
| MAC | FHKEY | RSSI | SNR |
| NMAC1 | NUM1 | 100.5 | 55 |
| NMAC2 | NUM2 | 45.4 | 45 |
| NMAC3 | NUM3 | 65.7 | 25 |
| NMAC4 | NUM4 | 545.5 | -8 |

1. MAC设备节点的MAC地址；
2. FHKEY：跳频种子；
3. RSSI：接收信号强度；
4. SNR：信噪比；
5. **时间参数1**
6. 时间范围：4s~14s；
7. 用于随机延时一段时间后重新请求绑定或重新请求入网；
8. **时间参数2**
9. 时间：1s；
10. 网关或设备节点发送数据包后等待的时间；
11. 超过该时间没有收到回复则认为是超时；
12. **时间参数3**
13. 时间：60s；
14. 设备节点超过这个没有收到轮询数据包则认为自身离线；
15. **入网上限数**
16. 数量：16个；
17. **Flag**

|  |  |  |
| --- | --- | --- |
| 4bit | 3bit | 1bit |
| 【协议版本】 | 【数据包类型】 | 【数据包方向】 |

1. 数据包类型

|  |  |
| --- | --- |
| 序号 | 类型 |
| 0 | 绑定包 |
| 1 | 入网包 |
| 2 | 轮询包 |
| 3 | 快速数据包 |

1. 数据包方向

|  |  |
| --- | --- |
| 序号 | 方向 |
| 0 | 节点向网关 |
| 1 | 网关向节点 |

1. 协议版本

1.0.0

1. **待续**

**（PS：文中所有数据包使用大端（Big-endian）类型）**